

Managing FileVault 2 with fdesetup on OS X Mountain Lion

Rich Trouton

Howard Hughes Medical Institute, Janelia Farm
Research Campus
Lead Help Desk Technician

Before we get started, there's two things I'd like to mention. The first is that, all of the slides, speakers' notes and the demos are available for download and I'll be providing a link at the end of the talk. I tend to be one of those folks who can't keep up with the speaker and take notes at the same time, so for those folks in the same boat, no need to take notes. Everything I'm covering is going to be available for download.

The second is to please hold all questions until afterwards. If you've got questions, make a note of them and ask me at the end. With luck, I'll be able to answer most of your questions during the talk itself.

FileVault 2 Under The Hood



To better understand the capabilities of `fdsetup` and how FileVault management works on 10.8, let's take a look underneath FileVault 2's hood to see how it handles authentication, unlocking and decryption.

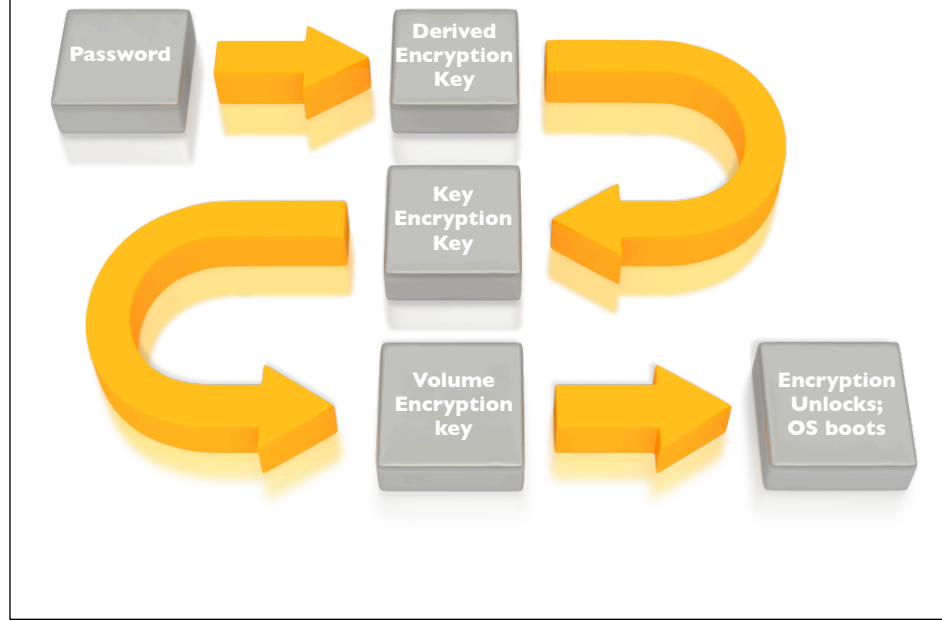
Keys Used By FileVault 2



- › Derived Encryption Key
- › Key Encryption Key
- › Volume Encryption Key

To begin with, passwords are almost irrelevant to FileVault 2's encryption. Instead, the system relies on a series of cryptographic keys granting access to two other layers of keys. These keys are the derived encryption key, the key encryption key and the volume encryption key.

Key-Driven Unlock Process



To give everyone an idea of how the keys are interacting with each other, here's a visual representation of what's happening when you log in at the pre-boot login screen.

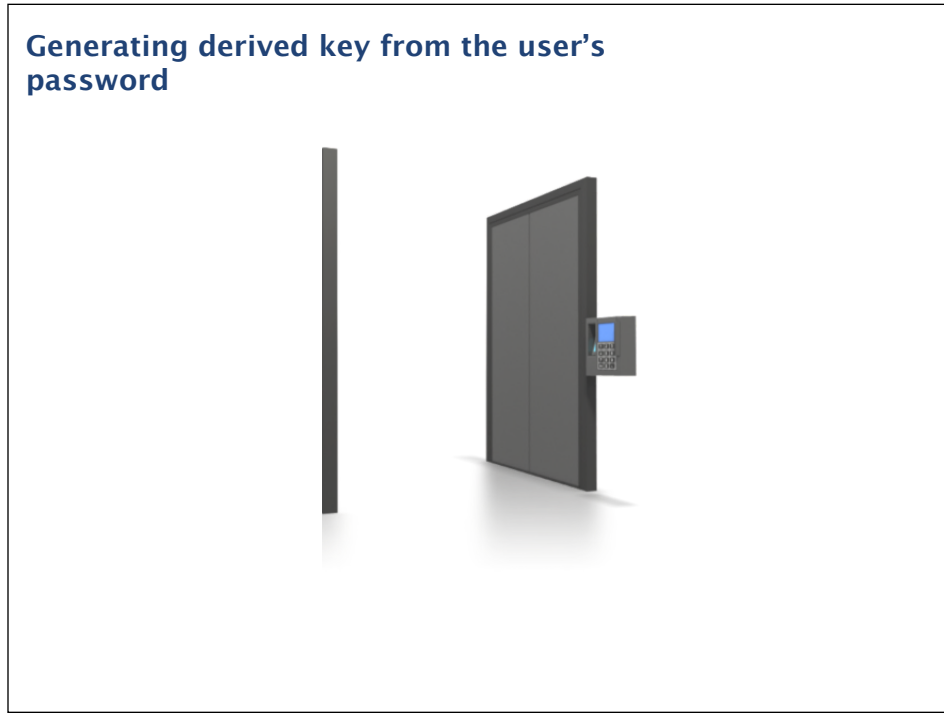
To take them from the bottom layer up, let's first look at the Volume Encryption Key. This is the key that is interacting with the CoreStorage volume that the FileVault 2 encryption process has created. All cryptographic operations on an encrypted CoreStorage volume are unique to that volume because a different volume encryption key is randomly generated for each volume. This is the key that is actually unlocking the encrypted volume and it's also the key that's deleted when a wipe command is sent to a FileVault 2 encrypted Mac.

On the next level up, there's the key encryption key. This key is generated when FileVault 2 encryption is initialized on a particular volume. It is used to unlock the volume encryption key one layer down and acts as the middleman between the volume encryption key and the derived keys. This middle layer allows the derived keys to change without affecting the derived keys ability to unlock the encrypted volume.

On the top layer, there's the derived encryption keys. These keys begin the chain-reaction of unlocking the other keys below it, resulting in the unlocking or decryption of the encrypted volume. Any derived key can be independently changed without affecting its ability to unlock the other two layers of keys.

Any given CoreStorage volume must support multiple cryptographic users, each with their own derived key. This is important because it means that there can be multiple ways to access the encrypted volume. In the case of FileVault 2's encryption, it means that multiple user accounts can be enabled to unlock an encrypted Mac at the pre-boot login screen. Derived keys are also used for the FileVault 2 recovery keys.

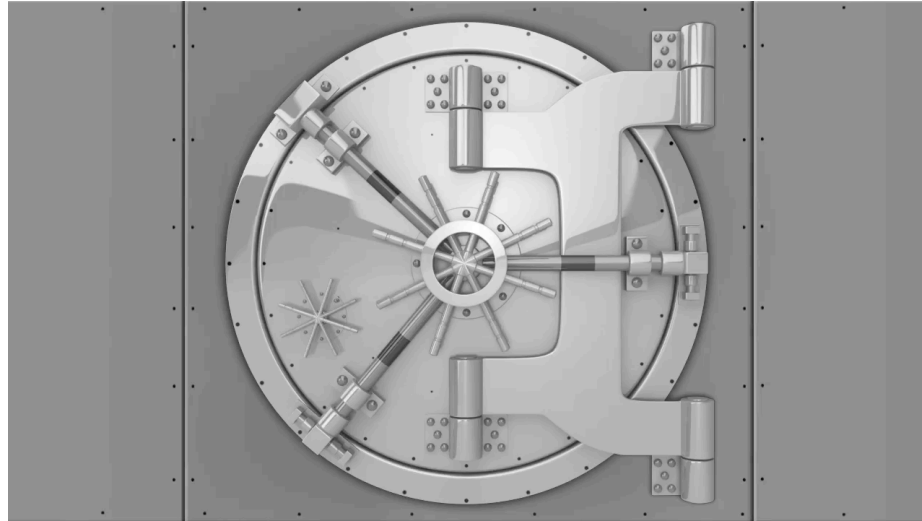
Generating derived key from the user's password



To break it down further:

In this illustration, this process opens the door you're seeing above. You enter your password and the password is converted to a derived key with the RSA Password-Based Key Derivation Function (otherwise known as PBKDF2).

Key validation and volume unlock



The derived key unlocks the key encryption key, represented here by the vault door. Once the key encryption key has been unlocked, it grants access to the volume encryption key.

The volume encryption key, represented here by the lock on the house, then unlocks the kernel and the OS boots.

The key encryption key is the key that gets updated when accounts are added, deleted or when passwords changed. With each account change, addition or removal, the key encryption key gets re-wrapped to allow for the updated information.

Pre-encryption access



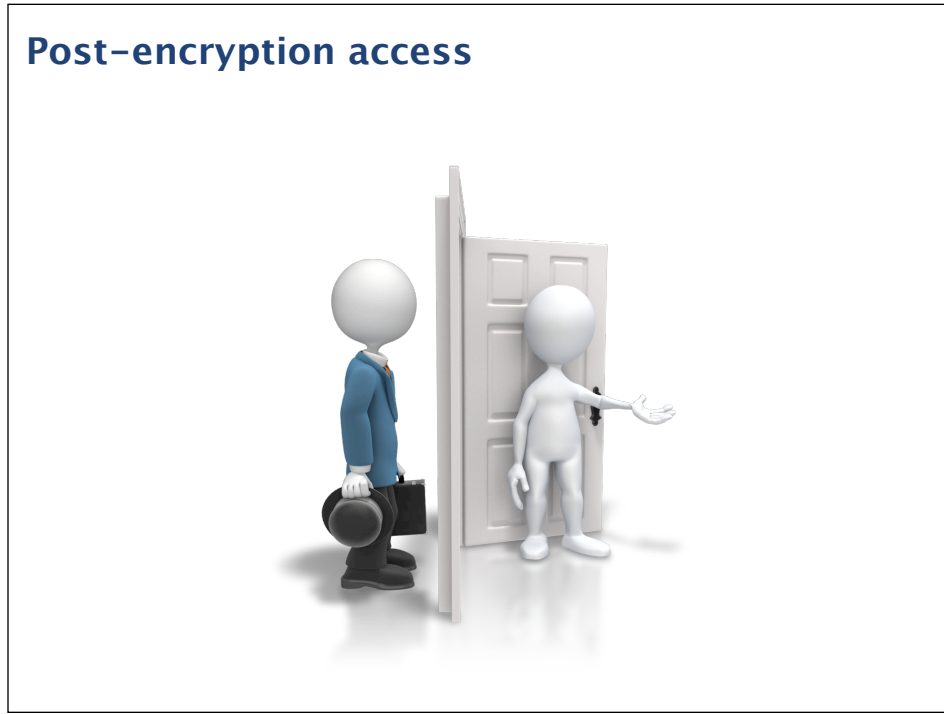
So how do you get a derived key in the first place? There's a couple of ways. The first is to have your access set up when FileVault 2 encryption is initialized. Since there are no pre-existing keys at this point, your password or other means of access get enabled at the same time that the encryption is initialized.

Post-encryption access



Once the encryption is turned on though, it's much tougher. The bouncer is standing by the door and he'll stop you from coming in unless you can properly show that you're legit. The only way to get in at this point is to have a friend who's already on the inside vouch for you.

Post-encryption access

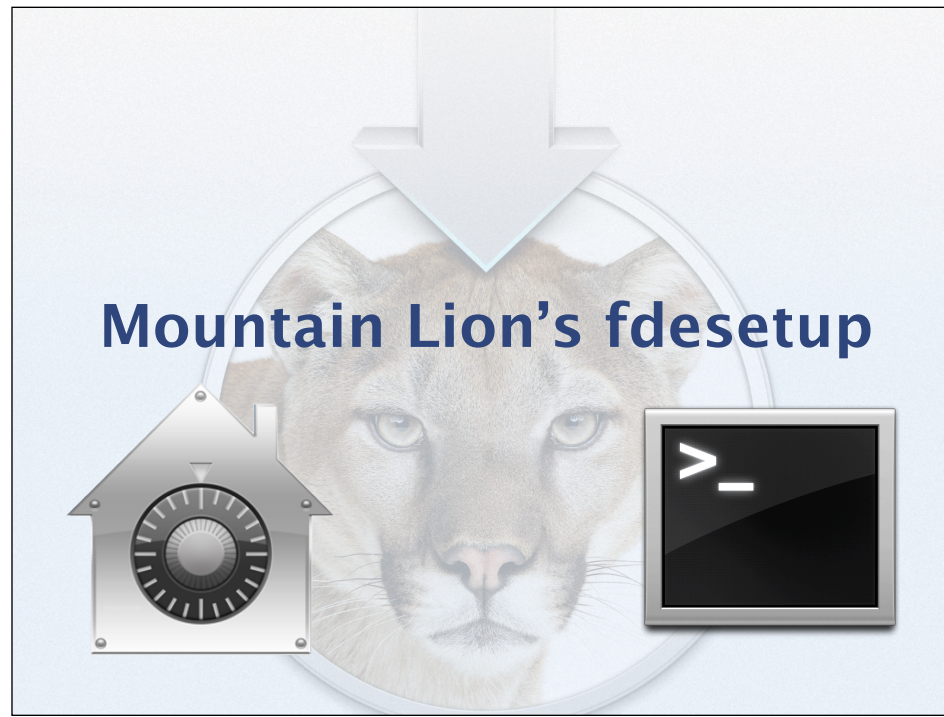


An existing derived key can be used to enable new derived keys, so your friend who's already inside can use his derived key to enable your account. At that point, a new derived key is generated to give you access and you're good to go.

No key? No access.



Without an existing derived key available, there's no way to get a new derived key set up. To make a long story short, if you don't have a friend on the inside, you're not getting in.



Before we dive into fdesetup on Mountain Lion, let's take a look at what FileVault 2 in 10.7 does not have.

You can monitor, unlock or decrypt a FileVault 2–encrypted boot drive using command line tools, but you can't start the encryption process from the command line using Apple's native tools. Instead, the encryption needs to be enabled from System Preference's FileVault preference pane.

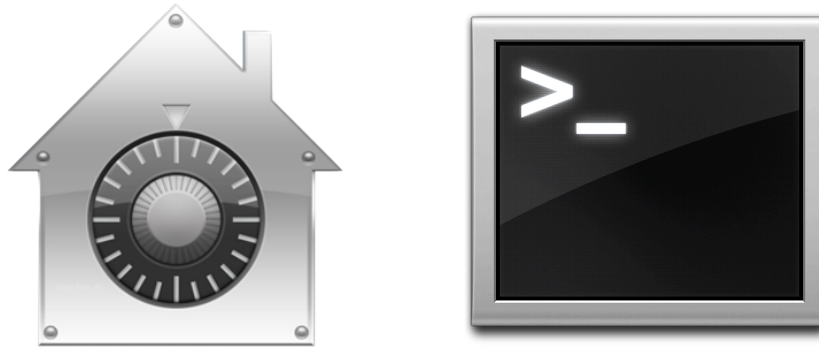
It is not possible to see who has FileVault 2–enabled accounts without looking at the pre–boot login screen.

It can be difficult to enable an account without using the FileVault preference pane.

It is not possible to remove an account from the list of enabled accounts without deleting the account or setting the account password to be blank.

You have to choose between using the individual alphanumeric recovery key or using the institutional recovery key using FileVaultMaster.keychain.

fdsetup overview



fdsetup on 10.8 allows FileVault 2 administration from the command line and solves all of those problems with its various functions. It will turn on FileVault 2 encryption using a variety of options, disable encryption, allow addition and removal of FileVault 2 enabled users from the command line, supply a current list of authorized users, provide encryption status and much more.

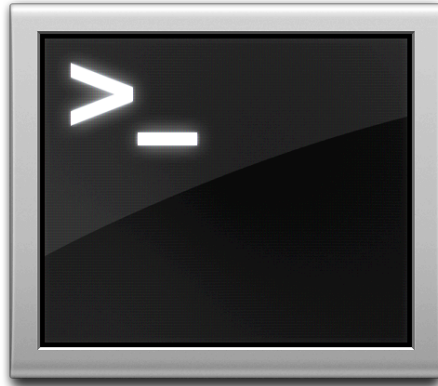
fdsetup commands



- › fdsetup enable
- › fdsetup disable
- › fdsetup add
- › fdsetup list
- › fdsetup remove
- › fdsetup sync

fdsetup has a number of verbs associated with it. The ones that may be most commonly used are enable, disable, add, list, remove and sync.

fdsetup enable



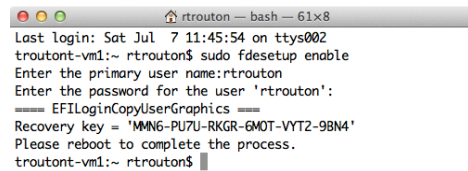
› Activates FileVault 2 Encryption

- Can set FileVault 2 encryption to use:
 - Individual alphanumeric recovery key
 - Institutional recovery key using FileVaultMaster.keychain
 - Both kinds of recovery key simultaneously
- Can enable multiple user accounts at time of encryption activation
- Can import user and certificate information

fdsetup is amazingly flexible when it comes to enabling FileVault 2 encryption from the command-line.

fdsetup enable

sudo fdsetup enable

A terminal window titled 'rtrouton - bash -- 61x8' showing the execution of the 'fdsetup enable' command. The output includes the last login time, the command being run, prompts for the primary user name and password, the EFI login copy user graphics, and a recovery key: 'MMN6-PU7U-RKGR-6MOT-VYT2-98M4'. The user is prompted to reboot to complete the process.

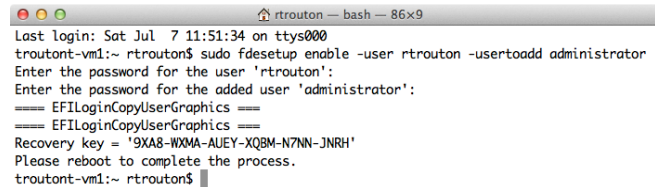
```
rtrouton - bash -- 61x8
Last login: Sat Jul 7 11:45:54 on ttys002
trouton-vm1:~ rtrouton$ sudo fdsetup enable
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
===== EFILoginCopyUserGraphics =====
Recovery key = 'MMN6-PU7U-RKGR-6MOT-VYT2-98M4'
Please reboot to complete the process.
trouton-vm1:~ rtrouton$
```

To start with the simplest method, you would run the command shown on the screen to enable FileVault 2 encryption. Next, you'll be prompted for the username and password of the primary user, which is the account you want to have appear at the FileVault 2 pre-boot login screen once the encryption is turned on. If everything's working properly, you'll next be given an alphanumeric individual recovery key and prompted to restart.

One thing that's very important to know is that the individual recovery key is not saved anywhere. You will need to make a record of it when it's displayed or you will not have it later.

fdsetup enable -user

sudo fdsetup enable -user username -usertoadd username



```
troutont-vm1:~ rtrouton$ sudo fdsetup enable -user rtrouton -usertoadd administrator
Enter the password for the user 'rtrouton':
Enter the password for the added user 'administrator':
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
Recovery key = '9XA8-WXMA-AUEY-XQBM-N7NN-JNRH'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

You can also enable additional user accounts at the time of encryption, as long as the accounts are either local or mobile network accounts. You would run the command as shown on the screen and specify the accounts you want. As part of this, you will be prompted for the account passwords.

After that, you'll be given an individual recovery key and prompted to restart. All of the accounts specified should appear at the FileVault 2 pre-boot login screen.

fdesetup enable -inputplist

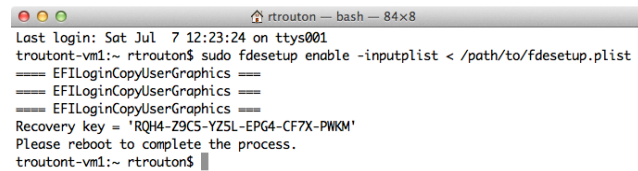
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>localadmin</string>
    <key>Password</key>
    <string>password</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>tom</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>harry</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

For those who want to automate the process, fdesetup also supports importing a property list file via standard input (stdin). The plist file needs to follow the format shown up on the screen and more users can be added by appending their information under the AdditionalUsers plist key.

fdsetup enable -inputplist

sudo fdsetup enable -inputplist < plistfile.plist



```
trouton -- bash -- 84x8
Last login: Sat Jul 7 12:23:24 on ttys001
troutont-vm1:~ rtrouton$ sudo fdsetup enable -inputplist < /path/to/fdsetup.plist
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
==== EFILoginCopyUserGraphics ====
Recovery key = 'RQH4-Z9C5-YZ5L-EPG4-CF7X-PWKM'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

Once the plist has been set up, you would run the command shown on the screen to enable FileVault 2 encryption and reference the information in the plist file.

Since the accounts and passwords are in the plist file, fdsetup does not need to prompt for passwords. Instead, the individual recovery key is displayed and the user is prompted to restart. All of the accounts specified in the plist file should appear at the FileVault 2 pre-boot login screen.

fdesetup enable -defer

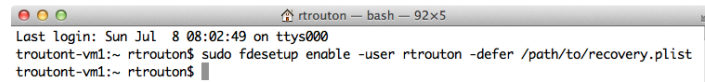
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnabledDate</key>
  <string>2012-07-08 08:05:54 -0400</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C293DEFEC</string>
  <key>LVGUID</key>
  <string>B7990442-91D1-4F5E-8F04-DDAC783610F2</string>
  <key>LVUID</key>
  <string>FF837400-B781-496E-8992-D5A11B5A1139</string>
  <key>PVUID</key>
  <string>E0FD9F00-5D41-4597-A108-73F0A463CC65</string>
  <key>RecoveryKey</key>
  <string>8MGZ-C7BL-ML2M-J6B4-HN35-ZDOA</string>
  <key>SerialNumber</key>
  <string>VMWk2fm2om0q0/em3zWslr2g</string>
</dict>
</plist>
```

To avoid the need to enter a password, `fdesetup` also has a `defer` flag that can be used with the `enable` verb to delay enabling FileVault 2 until after the user logs out. With the `defer` flag, the user will be prompted for their password at their next logout. The recovery key information is not generated until the user password is obtained, so the `defer` option requires a file location where this information will be written to as a plist file.

The plist file will be created as a root-only readable file and contain information similar to what's shown on the screen. For security reasons, this plist file should not stay on the encrypted system. It should be copied to a safe location and then securely deleted from the system.

fdsetup enable -defer

sudo fdsetup enable -user username -defer plistfile.plist

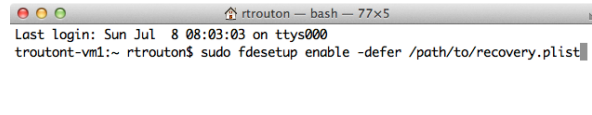
A terminal window with a title bar that reads "rtrouton — bash — 92x5". The terminal content shows a login message: "Last login: Sun Jul 8 08:02:49 on ttys000". Below that, the prompt "troutont-vm1:~ rtrouton\$" is followed by the command "sudo fdsetup enable -user rtrouton -defer /path/to/recovery.plist". The prompt then changes to "troutont-vm1:~ rtrouton\$" with a cursor at the end.

```
troutont-vm1:~ rtrouton$ sudo fdsetup enable -user rtrouton -defer /path/to/recovery.plist
troutont-vm1:~ rtrouton$
```

If you have a particular user account that you want to enable, you would run the command shown on the screen to defer enabling FileVault 2 and specify the account you want.

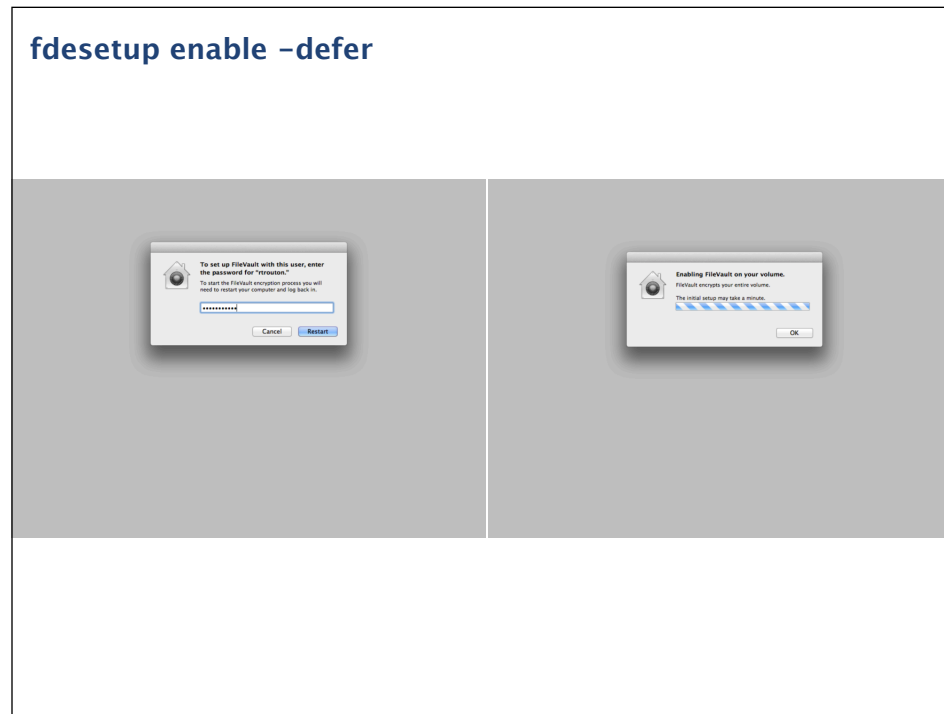
fdsetup enable -defer

sudo fdsetup enable -defer plistfile.plist

A terminal window with a title bar that reads "rtrouton — bash — 77x5". The terminal content shows a login message: "Last login: Sun Jul 8 08:03:03 on ttys000". Below that, the prompt "troutont-vm1:~ rtrouton\$" is followed by the command "sudo fdsetup enable -defer /path/to/recovery.plist" which is currently being typed.

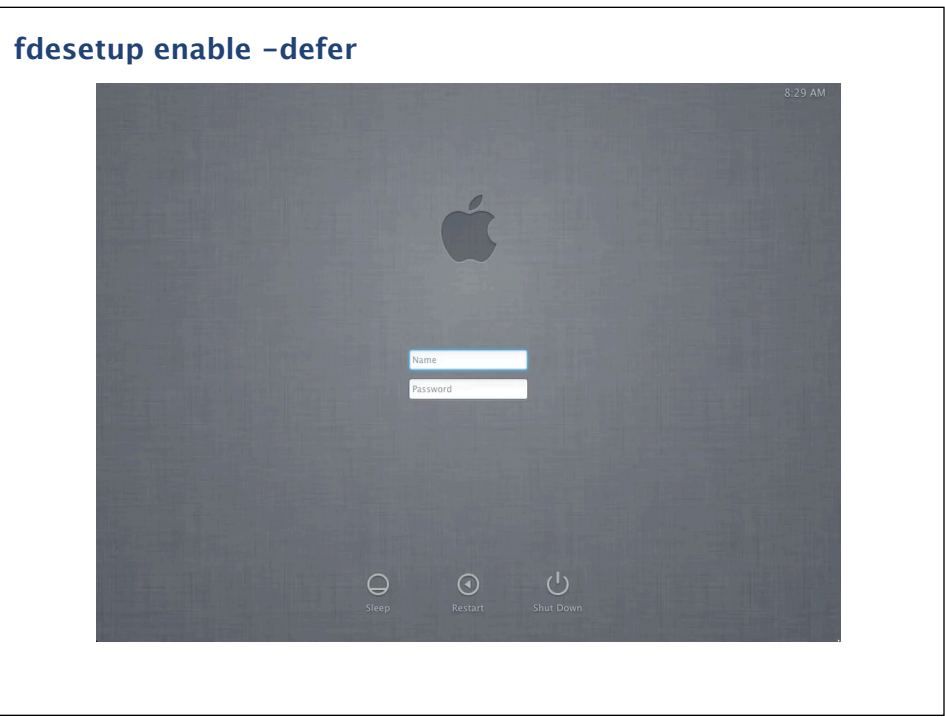
```
rtrouton — bash — 77x5
Last login: Sun Jul 8 08:03:03 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -defer /path/to/recovery.plist
```

If you don't want to specify the account, you would use the command shown on the screen. If there is no account specified, then the current logged-in user will be enabled for FileVault 2. If there is no user specified and no users are logged in when the command is run, then the next user that logs in will be chosen and enabled.



On logout, the user will be prompted to enter their account password. Once entered, FileVault 2 will be enabled and the recovery information plist file will be created. Once the enabling process is complete, the Mac will restart.

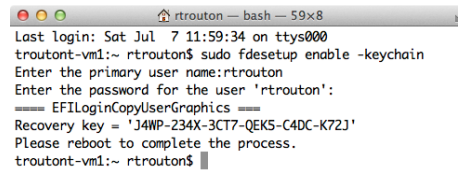
An important thing to keep in mind about the defer option is that it enables one single user account at the time of turning on FileVault 2 encryption. The defer option does not enable multiple user accounts and cannot be used to enable accounts once FileVault 2 encryption has been turned on.



Since this can be something that's better shown than explained, let's take a look at how the defer process works.

`fdsetup enable -keychain`

`sudo fdsetup enable -keychain`

A terminal window titled 'rtrouton -- bash -- 59x8' showing the execution of the command 'sudo fdsetup enable -keychain'. The output includes the user name 'rtrouton', a recovery key 'J4WP-234X-3CT7-QEKS-C4DC-K72J', and a message to reboot to complete the process.

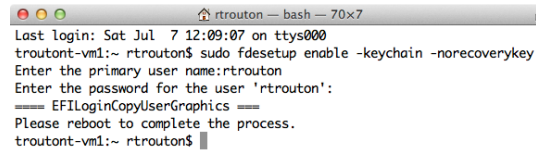
```
rtrouton-vm1:~ rtrouton$ sudo fdsetup enable -keychain
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Recovery key = 'J4WP-234X-3CT7-QEKS-C4DC-K72J'
Please reboot to complete the process.
rtrouton-vm1:~ rtrouton$
```

Another new capability of FileVault 2 in Mountain Lion is the ability to use the alphanumeric individual recovery key, an institutional recovery key using FileVaultMaster.keychain, or both kinds of recovery key at the same time.

As seen in the earlier examples, `fdsetup` will provide the individual recovery key by default. To use the institutional recovery key, the `-keychain` flag needs to be used as shown on the screen. The individual recovery key is displayed, but the encryption will also use the FileVaultMaster.keychain institutional recovery key. In case recovery is needed, either recovery key will work to unlock or decrypt the encrypted drive.

fdsetup enable -keychain

sudo fdsetup enable -keychain -norecoverykey

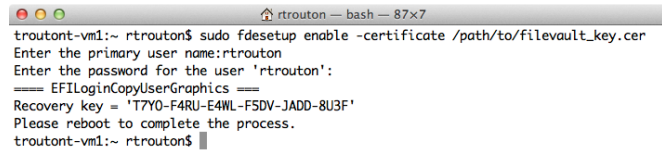
A terminal window titled 'rtrouton -- bash -- 70x7' showing the execution of the command 'sudo fdsetup enable -keychain -norecoverykey'. The output includes the last login time, the command being run, prompts for the primary user name and password, and a message indicating that the process will complete after a reboot.

```
rtrouton -- bash -- 70x7
Last login: Sat Jul 7 12:09:07 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -keychain -norecoverykey
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
===== EFILoginCopyUserGraphics =====
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

If you want to specify that only the FileVaultMaster keychain be used, both the `-keychain` and `-norecoverykey` flags need to be used when enabling encryption

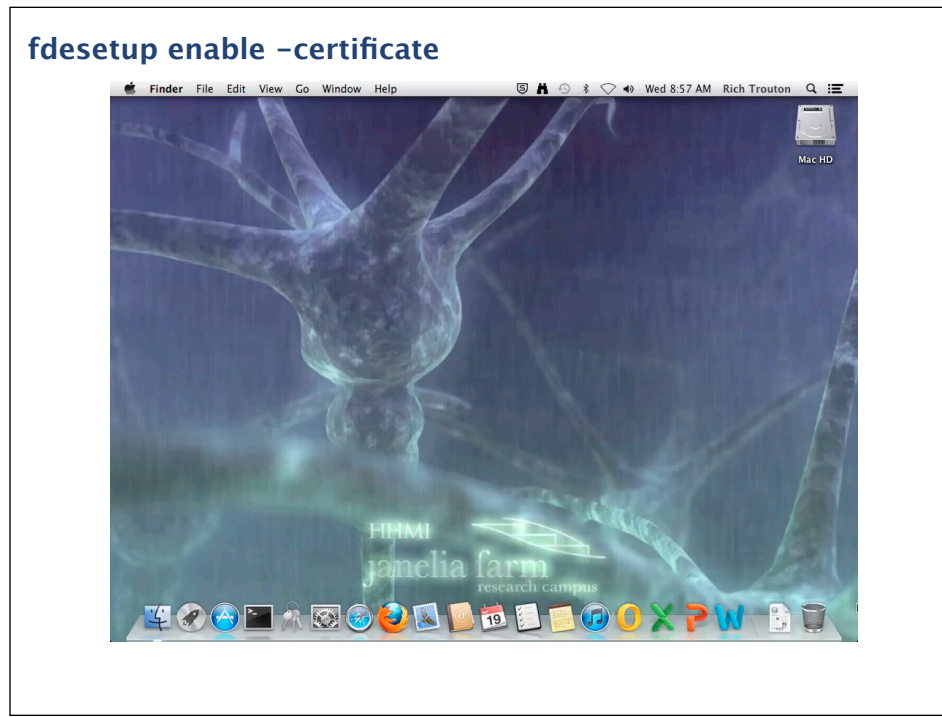
fdsetup enable -certificate

sudo fdsetup enable -certificate cert.cer

A terminal window titled 'rtrouton - bash - 87x7' showing the execution of the command 'sudo fdsetup enable -certificate /path/to/filevault_key.cer'. The output includes prompts for the primary user name and password, a separator line '==== EFILoginCopyUserGraphics ====', and the resulting recovery key 'T7Y0-F4RU-E4WL-F5DV-JADD-8U3F'. A message at the end states 'Please reboot to complete the process.'

```
troutont-vm1:~ rtrouton$ sudo fdsetup enable -certificate /path/to/filevault_key.cer
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Recovery key = 'T7Y0-F4RU-E4WL-F5DV-JADD-8U3F'
Please reboot to complete the process.
troutont-vm1:~ rtrouton$
```

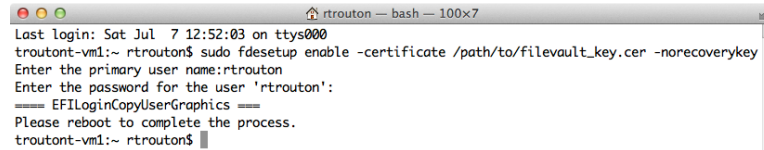
fdsetup is also capable of creating a FileVaultMaster keychain and automatically storing it in /Library/Keychains. To do this, an existing FileVault 2 public key needs to be available as a DER encoded certificate file. Once that's available, the command shown on the screen will enable FileVault 2, automatically create the institutional recovery key with the supplied certificate file and store it as /Library/Keychains/FileVaultMaster.keychain



Let's take a look at how you would create a DER encoded certificate file from an existing public key.

fdsetup enable -certificate

sudo fdsetup enable -certificate cert.cer -norecoverykey

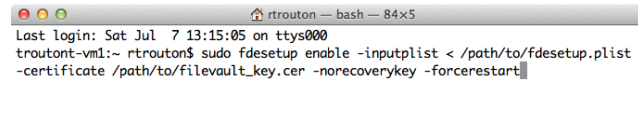


```
trouton -- bash -- 100x7
Last login: Sat Jul 7 12:52:03 on ttys000
trouton-vm1:~ rtrouton$ sudo fdsetup enable -certificate /path/to/filevault_key.cer -norecoverykey
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
==== EFILoginCopyUserGraphics ====
Please reboot to complete the process.
trouton-vm1:~ rtrouton$
```

To specify that only the FileVaultMaster keychain be used as the recovery key, you would add the `norecoverykey` flag to the command.

fdsetup enable -forcerestart

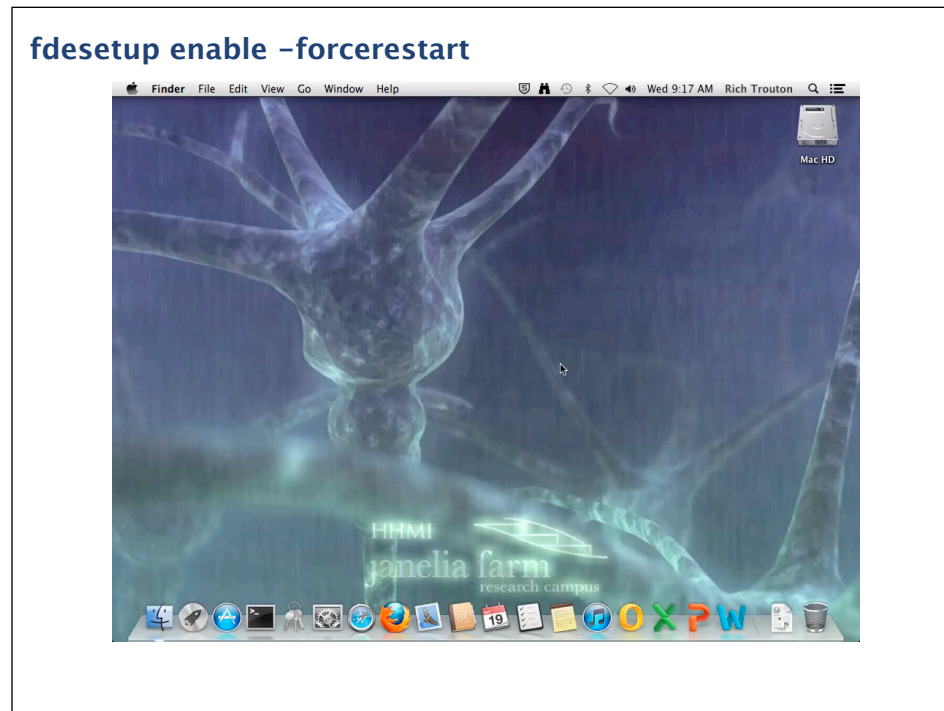
**sudo fdsetup enable -inputplist < plistfile.plist-certificate
cert.cer -norecoverykey
-forcerestart**

A terminal window titled "rtrouton - bash - 84x5" showing the execution of the command. The output includes the last login time and the command being run.

```
rtrouton - bash - 84x5
Last login: Sat Jul 7 13:15:05 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup enable -inputplist < /path/to/fdsetup.plist
-certificate /path/to/filevault_key.cer -norecoverykey -forcerestart
```

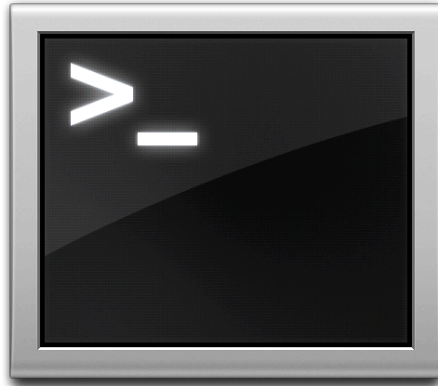
Along with the various options for enabling, it's also possible to force a restart of the Mac once FileVault 2 has been successfully configured. This can help automate the process of enabling FileVault 2 on a Mac if no input from a logged-in user is needed.

For example, an organization may want to pre-configure its Macs to automatically encrypt with FileVault 2 at first boot with a local admin account enabled. It also wants to use only the institutional recovery key. If a plist with the desired account information and a certificate file to create the institutional recovery key is available, the command shown on the screen could be run to enable FileVault 2 and force a restart at the first boot.



Since this combines three different enable options, let's take a look at how it works when you run that command to automatically encrypt. In this case, I'm going to be enabling three accounts via a plist file and setting the institutional key as the sole recovery key.

`fdesetup disable`



› Disables FileVault 2 encryption

In contrast to all of the various options available for enabling FileVault 2 using `fdesetup`, the command to turn off FileVault 2 encryption is `fdesetup disable`. There are no additional flags associated with this command.

`fdsetup add`



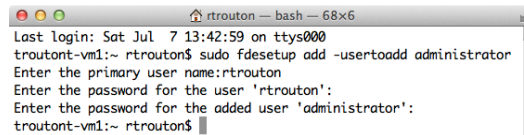
› Enables additional accounts after FileVault 2 encryption is complete

- Can enable multiple user accounts
- Can import user information

Once the Mac has been fully encrypted with FileVault 2, you can add additional users using `fdsetup`. To do so, you will need to provide both the username and password of a previously enabled account as well as the password of the account you want to add.

fdsetup add -usertoadd

sudo fdsetup add -usertoadd username

A terminal window titled 'rtrouton - bash - 68x6' showing the execution of the 'fdsetup' command. The output shows the user 'rtrouton' being prompted for their name and password, and then the user 'administrator' being added.

```
rtrouton - bash - 68x6
Last login: Sat Jul 7 13:42:59 on ttys000
troutont-vm1:~ rtrouton$ sudo fdsetup add -usertoadd administrator
Enter the primary user name:rtrouton
Enter the password for the user 'rtrouton':
Enter the password for the added user 'administrator':
troutont-vm1:~ rtrouton$
```

The command shown on the screen will enable a specified user on this encrypted Mac. The primary user can be any account on the Mac that's already been enabled for use with FileVault 2.

fdsetup add -inputplist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>rtrouton</string>
    <key>Password</key>
    <string>password</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>fcheeryble</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>nickleby</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

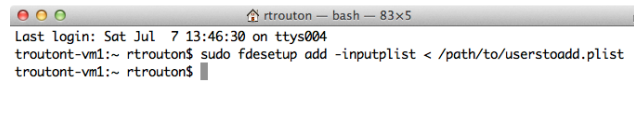
For those who want to automate the process, `fdsetup` also supports importing a plist file via standard input (stdin). The plist needs to follow the format shown up on the screen.

When adding additional users using a plist file, the top level Username key is ignored, and the Password key value should be an existing FileVault user's password. More users can be added by appending their information under the AdditionalUsers plist key.

The `fdsetup` man page references the ability to use the recovery key to add additional users. However, this function does not work as of 10.8.3. Apple is aware of this issue and is planning to fix it in a future OS release.

fdsetup add -inputplist

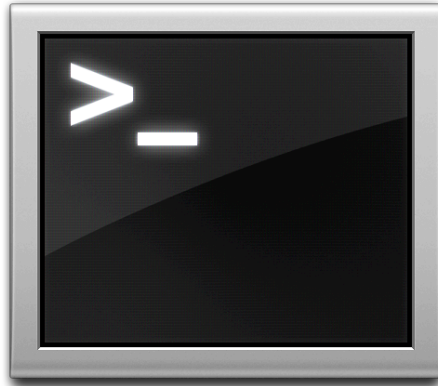
sudo fdsetup add -inputplist /path/to/plistname.plist

A terminal window titled "rtrouton - bash - 83x5" showing the execution of the command. The window content is: "Last login: Sat Jul 7 13:46:30 on ttys004", "troutont-vm1:~ rtrouton\$ sudo fdsetup add -inputplist < /path/to/userstoadd.plist", and "troutont-vm1:~ rtrouton\$".

```
rtrouton - bash - 83x5
Last login: Sat Jul 7 13:46:30 on ttys004
troutont-vm1:~ rtrouton$ sudo fdsetup add -inputplist < /path/to/userstoadd.plist
troutont-vm1:~ rtrouton$
```

Once the plist has been set up, you can run the command shown on the screen to add additional users by referencing the account information in the plist file.

fdsetup list



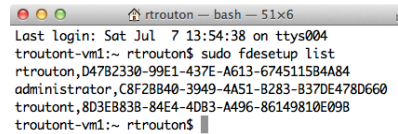
› Displays enabled accounts

- List includes the accounts' usernames and UIDs

To list all accounts enabled for FileVault 2, fdsetup includes the list verb.

fdsetup list

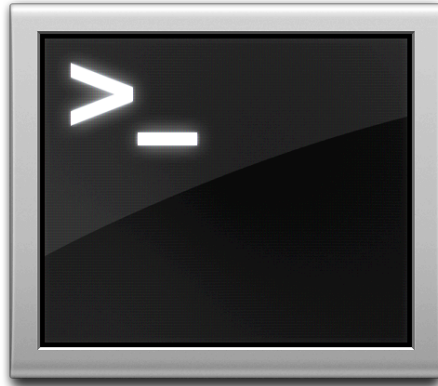
sudo fdsetup list



```
rttrouton -- bash -- 51x6
Last login: Sat Jul 7 13:54:38 on ttys004
troutont-vm1:~ rttrouton$ sudo fdsetup list
rttrouton,D47B2330-99E1-437E-A613-674511584A84
administrator,C8F28B40-3949-4A51-8283-837DE478D660
troutont,8D3E883B-84E4-4DB3-A496-86149810E098
troutont-vm1:~ rttrouton$
```

To get a list of all FileVault 2 enabled accounts on your Mac, you would run the command shown on the screen. All enabled accounts will be listed with both the accounts' username and UUID.

`fdsetup remove`



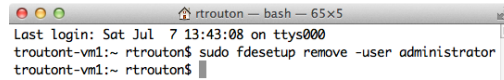
› Removes accounts from the list of FileVault 2 enabled accounts

- Can disable using account username
- Can disable using account UUID

To remove accounts from the list of FileVault 2 enabled accounts, `fdsetup` includes the `remove` verb. You can remove users by using either the username or the account's UUID.

fdesetup remove -user

sudo fdesetup remove -user username

A terminal window titled "rtrouton -- bash -- 65x5" showing the execution of the command "sudo fdesetup remove -user administrator". The output shows the last login time and the successful execution of the command.

```
rtrouton -- bash -- 65x5
Last login: Sat Jul 7 13:43:08 on ttys000
troutant-vm1:~ rtrouton$ sudo fdesetup remove -user administrator
troutant-vm1:~ rtrouton$
```

To remove the account by username, you would run the command as shown on the screen and provide the account's username.

fdesetup remove -uuid

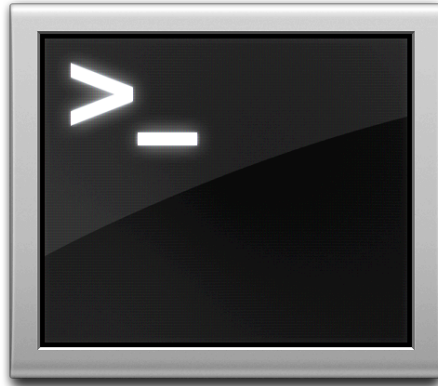
sudo fdesetup remove -uuid uuid_here

A terminal window screenshot showing the command execution. The window title is "rtrouton -- bash -- 89x5". The prompt is "troutont-vm1:~ rtrouton\$". The command entered is "sudo fdesetup remove -uuid C8F28B40-3949-4A51-B283-B37DE478D660". The prompt returns to "troutont-vm1:~ rtrouton\$".

```
troutont-vm1:~ rtrouton$ sudo fdesetup remove -uuid C8F28B40-3949-4A51-B283-B37DE478D660
troutont-vm1:~ rtrouton$
```

To remove the account using the UUID, you would run the command as shown on the screen and provide the account's UUID.

`fdsetup sync`

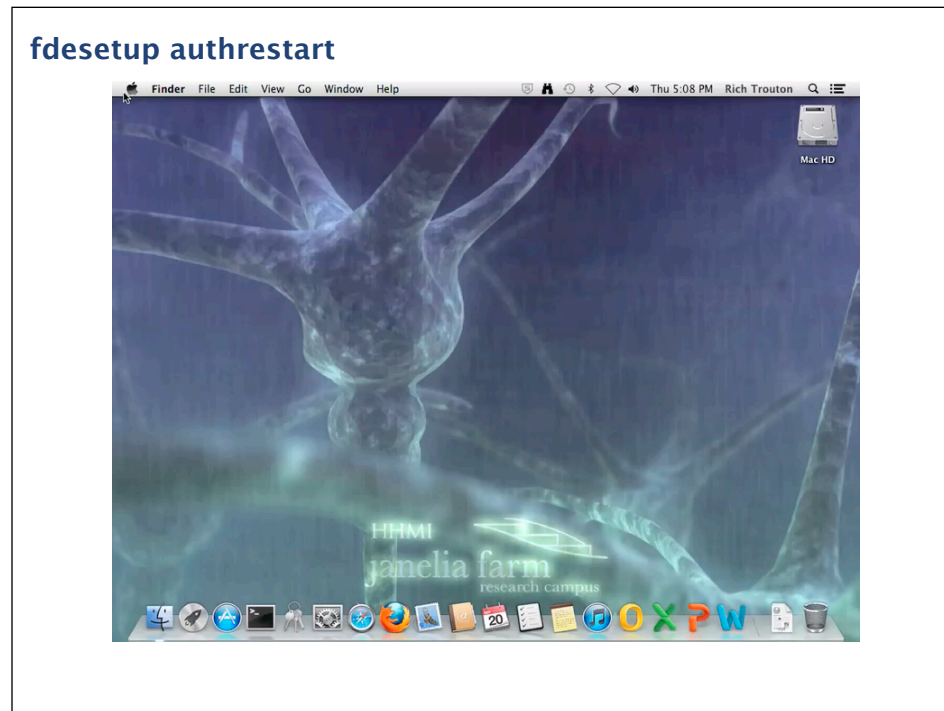


› Compares directory service account information with Mac's list of FileVault 2 enabled accounts

- Removes users that have been removed from the directory service.
- Does not add directory service accounts to list of FileVault 2 enabled accounts.

`fdsetup` also has the `sync` verb, which allows FileVault 2 to check with the Mac's directory service and see which accounts have been changed. Its main use currently is to automate the disabling of FileVault 2-enabled accounts by checking the directory service to see which accounts have been removed. If an account has been removed from the directory service, running `fdsetup sync` on an encrypted Mac will automatically remove the account from the list of FileVault 2 enabled accounts. The sync only affects the account's FileVault 2 status and will not remove the account or account home folder from the Mac.

One important thing to know is that `sync` does not allow accounts to be automatically added, only removed.







10.8.2 introduced a new function for `fdsetup`, `authrestart`. `fdsetup authrestart` will allow a one-time restart of a FileVault 2-encrypted Mac which goes to the regular login window instead of the FileVault 2 pre-boot login screen.

As this is something that's best shown, here's what happens when "`fdsetup authrestart`" is executed on a Mountain Lion Mac that's encrypted with FileVault 2 . When you run the command, it asks for a password or recovery key. The password must be an account that has been enabled for FileVault 2. After that, it puts an unlock key in system memory and reboots. On reboot, the reboot process automatically clears the unlock key from memory.

FileVault 2 Management Solutions Using fdesetup



There are a number of FileVault 2 management solutions that use `fdesetup` to manage FileVault 2 on Mountain Lion Macs, available from JAMF Software, Dell and open source projects.

	Name	Supported OSs	Recovery Key Support	Vendor
	Cauliflower Vest	10.7.x and 10.8.x	Individual	Open Source
	The Casper Suite	10.8.x	Individual and Institutional	JAMF Software
	Credant Enterprise Edition for Mac	10.7.x and 10.8.x	Institutional	Dell
	Crypt	10.7.x and 10.8.x	Individual	Open Source

The grid shows a listing of the management solutions that I've worked with. All have their strengths, so I recommend evaluating them carefully to find the one that meets your needs.

Of the ones listed, only JAMF Software opted to rely solely on `fdesetup` for its FileVault 2 management so I'm going to be using Casper as my example management suite for the rest of the talk.

Managing FileVault 2 with the Casper Suite



In Casper 8.6 and higher, there's a Disk Encryption Configurations option available in the Management settings. However, if you don't have rights to manage disk encryption, you won't see this option.

Managing FileVault 2 with the Casper Suite

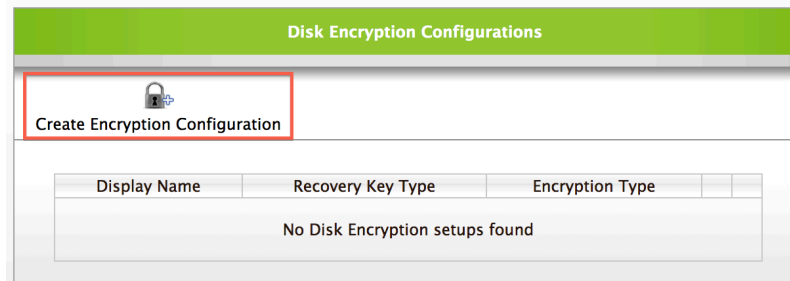
JSS - Inventory Tab Privileges	
<input checked="" type="checkbox"/>	View Inventory Tab
<input checked="" type="checkbox"/>	Perform Advanced Searches
<input checked="" type="checkbox"/>	Save Advanced Searches
<input checked="" type="checkbox"/>	View Saved Searches
<input checked="" type="checkbox"/>	Add Computers Manually
<input checked="" type="checkbox"/>	View Details on Inventory Items
<input checked="" type="checkbox"/>	View License Serial Numbers
<input checked="" type="checkbox"/>	Download Files Attached to Inventory Items
<input checked="" type="checkbox"/>	View Computer Logs
<input checked="" type="checkbox"/>	Edit Inventory Items
<input checked="" type="checkbox"/>	Edit Autorun Data
<input checked="" type="checkbox"/>	Send Email to End Users via JSS
<input checked="" type="checkbox"/>	Delete Inventory Items
<input checked="" type="checkbox"/>	Send Inventory Requests to Mobile Devices
<input checked="" type="checkbox"/>	Enroll Mobile Devices
<input checked="" type="checkbox"/>	Unmanage Mobile Devices
<input checked="" type="checkbox"/>	Enroll OS X Computers
<input type="checkbox"/>	View Disk Encryption Recovery Key

JSS - Management Tab Privileges	
<input checked="" type="checkbox"/>	View Management Tab
<input checked="" type="checkbox"/>	Manage Policies
<input checked="" type="checkbox"/>	Manage Configuration Profiles
<input checked="" type="checkbox"/>	Manage Computer Enrollment Preferences
<input checked="" type="checkbox"/>	Manage Developer Certificate Identities
<input type="checkbox"/>	Manage Disk Encryption
<input type="checkbox"/>	Manage Disk Encryption Institutional Key
<input checked="" type="checkbox"/>	Manage Managed Preferences
<input checked="" type="checkbox"/>	Manage PreStages
<input checked="" type="checkbox"/>	Manage Restricted Software
<input checked="" type="checkbox"/>	Manage Smart Computer Groups
<input checked="" type="checkbox"/>	Manage Static Computer Groups
<input checked="" type="checkbox"/>	Manage Mobile Device Profiles
<input checked="" type="checkbox"/>	Manage Mobile Device Remote Commands
<input checked="" type="checkbox"/>	Send Mobile Device Remote Lock Command
<input checked="" type="checkbox"/>	Send Mobile Device Remote Passcode Command
<input checked="" type="checkbox"/>	Send Mobile Device Remote Wipe Command
<input checked="" type="checkbox"/>	Manage Mobile Device Enrollments
<input checked="" type="checkbox"/>	Manage Mobile Device App Catalog
<input checked="" type="checkbox"/>	Manage Mobile Device eBook Catalog
<input checked="" type="checkbox"/>	Manage Smart Mobile Device Groups
<input checked="" type="checkbox"/>	Manage Static Mobile Device Groups
<input checked="" type="checkbox"/>	Send Computer Remote Lock Command
<input checked="" type="checkbox"/>	Send Computer Remote Wipe Command
<input checked="" type="checkbox"/>	Send Computer Unmanage Command

Casper Remote Privileges	
<input checked="" type="checkbox"/>	Use Casper Remote
<input checked="" type="checkbox"/>	Install/Uninstall Software Remotely
<input checked="" type="checkbox"/>	Run Scripts Remotely
<input checked="" type="checkbox"/>	Map Printers Remotely
<input checked="" type="checkbox"/>	Add Dock Items Remotely
<input checked="" type="checkbox"/>	Manage Local User Accounts Remotely
<input checked="" type="checkbox"/>	Change Casper's SSH Accounts Remotely
<input checked="" type="checkbox"/>	Bind to Active Directory Remotely
<input checked="" type="checkbox"/>	Set Open Firmware/EFI Passwords Remotely
<input checked="" type="checkbox"/>	Reboot Computers Remotely
<input checked="" type="checkbox"/>	Perform Maintenance Tasks Remotely
<input checked="" type="checkbox"/>	Search for Files/Processes Remotely
<input type="checkbox"/>	Enable Disk Encryption Configurations Remotely

In order to access it and other disk encryption settings, some access rights need to be granted to your account on the Casper server. It's important to know that these rights are not granted automatically to existing admin accounts. They will need to be enabled on a per-account basis.

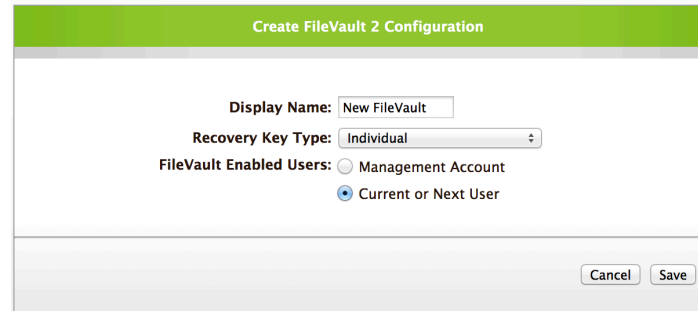
Managing FileVault 2 with the Casper Suite



If you haven't previously set up a disk encryption configuration, there won't be any existing setups found. To get started on creating one, you would click on the "Create Encryption Configuration" button.

Managing FileVault 2 with the Casper Suite

Everything in this window is an **fdesetup** option



The screenshot shows a dialog box titled "Create FileVault 2 Configuration". It contains the following fields and options:

- Display Name:
- Recovery Key Type:
- FileVault Enabled Users: Management Account, Current or Next User

At the bottom right, there are "Cancel" and "Save" buttons.

As mentioned previously, JAMF Software opted to rely solely on `fdesetup` for its FileVault 2 management. All Casper disk encryption configuration options use `fdesetup`'s capabilities.

Managing FileVault 2 with the Casper Suite

The image displays three screenshots of the 'Create FileVault 2 Configuration' dialog box, illustrating how different GUI options are translated into fdesetup commands.

Top Left Screenshot: Shows 'Recovery Key Type' set to 'Individual' and 'FileVault Enabled Users' set to 'Current or Next User'. The resulting command is `fdesetup enable -defer plistfile.plist`.

Top Right Screenshot: Shows 'Recovery Key Type' set to 'Individual And Institutional', 'Institutional Recovery Key' set to 'Upload', and 'FileVault Enabled Users' set to 'Management Account'. The resulting command is `fdesetup enable -user username -keychain`.

Bottom Screenshot: Shows 'Recovery Key Type' set to 'Individual' and 'FileVault Enabled Users' set to 'Management Account'. The resulting command is `fdesetup -enable -user username`.

To illustrate, let's take some of the options available and translate them into their fdesetup equivalents.

Managing recovery keys



Management of recovery keys is really important with FileVault 2-encrypted Macs, as they are your disaster recovery method. Disaster recovery is something you always should plan for when dealing with encrypted machines. After all, these are designed to protect your data against external threats unless properly authenticated. If your OS takes a dive, your normal way of unlocking may no longer work.

When the Mac has been encrypted using a Casper policy, Casper can provide back the recovery key if needed.

Managing recovery keys



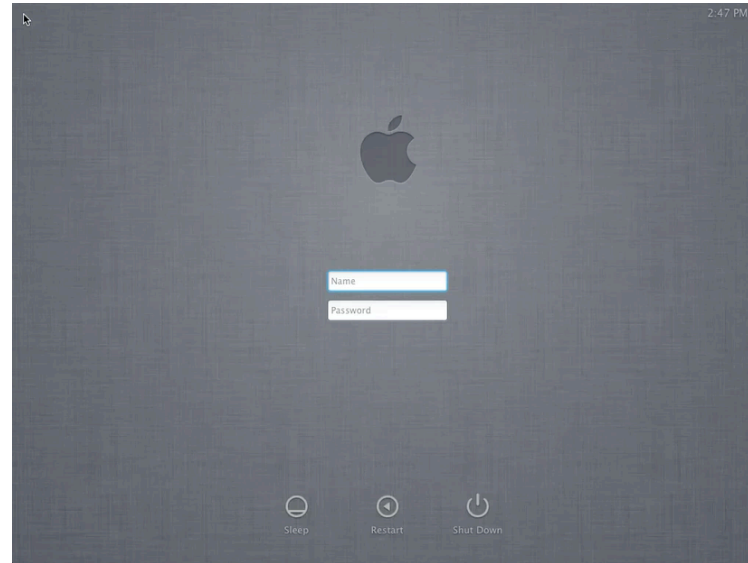
- › Individual Key
- › Institutional Key
- › Individual And Institutional Key

The three recovery key options available are:

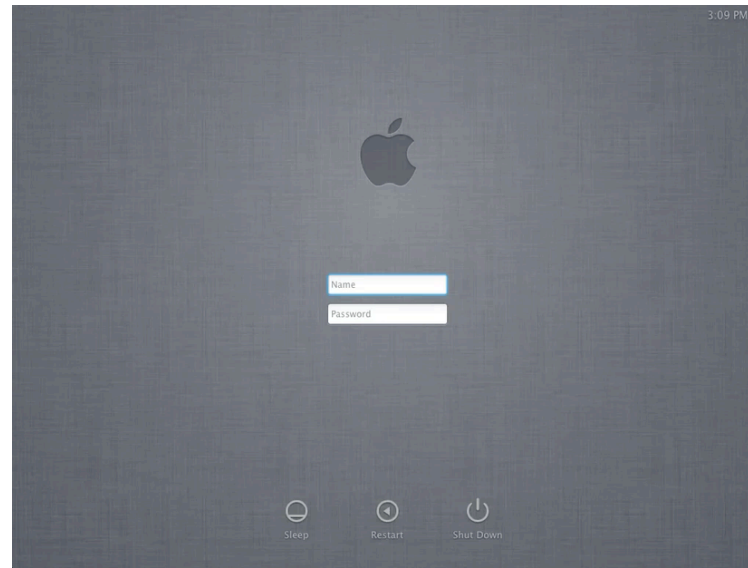
- A. The individual key, which is the alphanumeric key that FileVault 2 generates if there is not a FileVaultMaster keychain on the Mac.
- B. The institutional key, which is a pre-configured FileVaultMaster keychain
- C. The individual and institutional key, which is fdesetup's new way of using both the alphanumeric key and a FileVaultMaster keychain together on one machine. With this option, you can have two recovery key options available on one encrypted Mac.

The individual recovery key will be generated and sent up to the Casper server without additional work on the Casper admin's part. If an institutional key is used, that institutional key will need to be generated and uploaded to the Casper server.

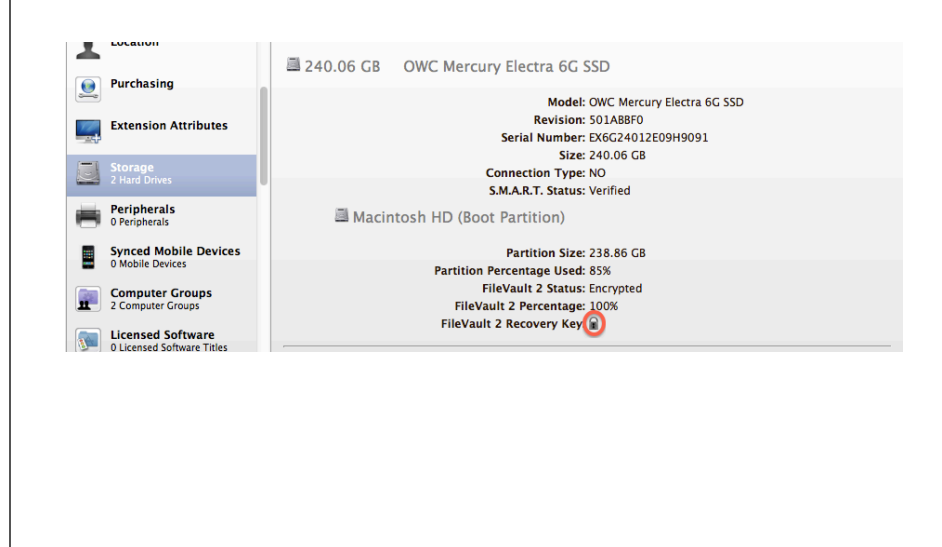
Uploading institutional keys



Downloading institutional keys

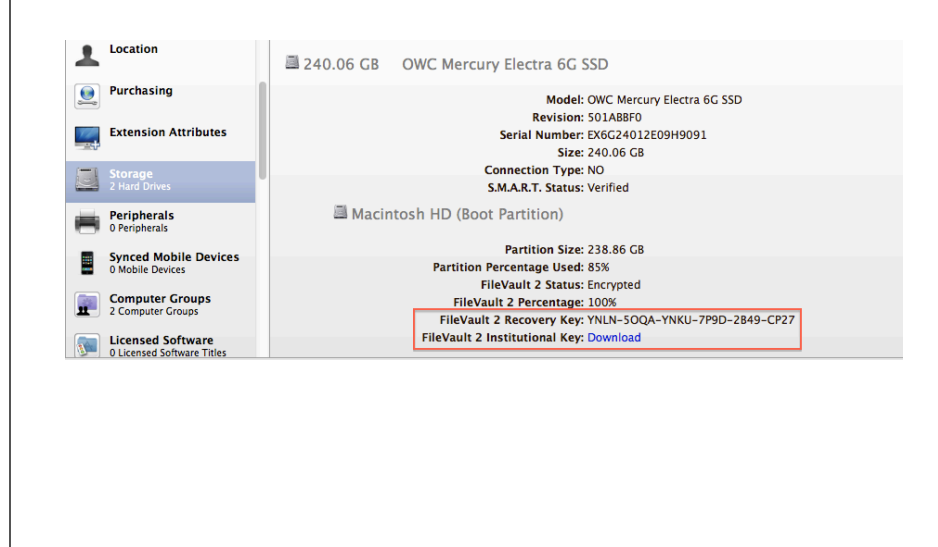


Accessing recovery keys



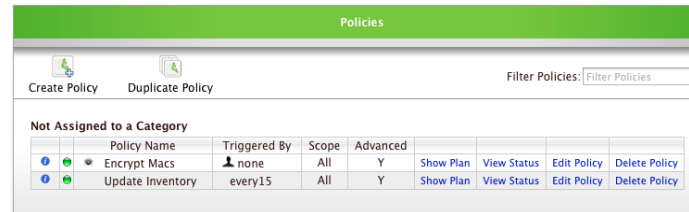
Once a Mac has been encrypted using a disk encryption policy, authorized accounts will be able to access the recovery key from that Mac's inventory listing. To access the recovery key, go to Storage and then click on the lock icon next to the FileVault 2 Recovery Key listing.

Accessing recovery keys



Once clicked, the icon disappears and reveals the recovery key information. In this case, the Mac in question is using both the individual key and the institutional recovery key together. Clicking the download link will give you the .p12 file you would need to build a complete recovery keychain.

Encryption Policies in Self Service



Policies

Create Policy Duplicate Policy Filter Policies:

Not Assigned to a Category

	Policy Name	Triggered By	Scope	Advanced				
<input type="checkbox"/>	Encrypt Macs	none	All	Y	Show Plan	View Status	Edit Policy	Delete Policy
<input type="checkbox"/>	Update Inventory	every15	All	Y	Show Plan	View Status	Edit Policy	Delete Policy

One way to encrypt your Macs is to do it with Self Service policies. Here's an example of how you may want to set one up.

Encryption Policies in Self Service

Edit Policy: Encrypt Macs

General Scope Self Service Packages Scripts Printers Dock Accounts Reboot Advanced

Display and Execution Settings

Display Name:

Category:

Triggered By:

Execution Frequency: Make Available Offline

► Date and Time Limitations

► Network Limitations

► Override Default Policy Settings

Cancel Save

You'll want to give it a descriptive name and set it so that it's triggered by Self Service. Execution frequency should be set to either Once Per Computer or Ongoing. I've got mine set here to Ongoing in the event that you want to allow for decryption, then re-encryption.

Encryption Policies in Self Service

The screenshot shows the 'Edit Policy: Encrypt Macs' window in Self Service. The window has a green header and a navigation bar with tabs: General, Scope, Self Service, Packages, Scripts, Printers, Dock, Accounts, Reboot, and Advanced. The 'Reboot' tab is selected.

Under the 'Reboot' tab, there are two sections:

- If Nobody Is Logged In:** Radio buttons for 'Do not Reboot' (selected), 'Reboot Immediately', and 'Reboot only if a package or SWU requires'.
- If Anybody Is Logged In:** Radio buttons for 'Do not Reboot' (selected), 'Reboot', 'Reboot only if a package or SWU requires', and 'Reboot Immediately'. Below these is a text field 'Give User 5 minutes after clicking OK'.

Below these sections is the **Reboot Options** section:

- Message:** A text area containing 'Log out at the next opportunity to start FileVault encryption'. A checkbox 'Display message if not rebooting' is checked.
- Reboot To:** A dropdown menu set to 'Current Startup Disk'.

At the bottom right, there are 'Cancel' and 'Save' buttons.

Here I'm setting it to not reboot because I'm planning to do a deferred enable, which requires the user to log out and supply their account's password.

Encryption Policies in Self Service

The screenshot shows the 'Edit Policy: Encrypt Macs' window with the following sections:

- Maintenance:** Includes checkboxes for 'Update Inventory' (checked), 'Update Prebindings', 'Flush System Caches', 'Reset Computer Names', 'Fix Permissions', 'Flush User Caches', 'Self Heal Packages', 'Fix ByHost Files', and 'Verify Startup Disk'.
- Files & Processes:** Includes search fields for file path, name, and process, along with checkboxes for 'Delete if found', 'Update Locate DB', and 'Kill if found'.
- Disk Encryption Configurations:** A table with columns for 'Enabled', 'Display Name', 'Recovery Key Type', and 'Encryption Type'. The 'Individual FileVault' row is selected.

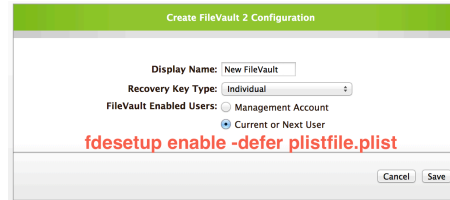
Enabled	Display Name	Recovery Key Type	Encryption Type
<input type="checkbox"/>	Individual and Institutional (both keys) FileVault	Individual And Institutional	FileVault 2
<input type="checkbox"/>	Individual and Institutional (public key only) FileVault	Individual And Institutional	FileVault 2
<input checked="" type="checkbox"/>	Individual FileVault	Individual	FileVault 2
<input type="checkbox"/>	Institutional (public key only) FileVault	Institutional	FileVault 2

Management Framework Options - Not required for clients running 7.3 or later

Buttons: Cancel, Save

Last but not least, I'm setting the disk encryption configuration I want to use and specifying that the Casper agent on the machine send an updated inventory back to the Casper server. In this case, I'm specifying that the recovery key type be the alphanumeric individual recovery key.

Encryption Policies in Self Service



Create FileVault 2 Configuration

Display Name:

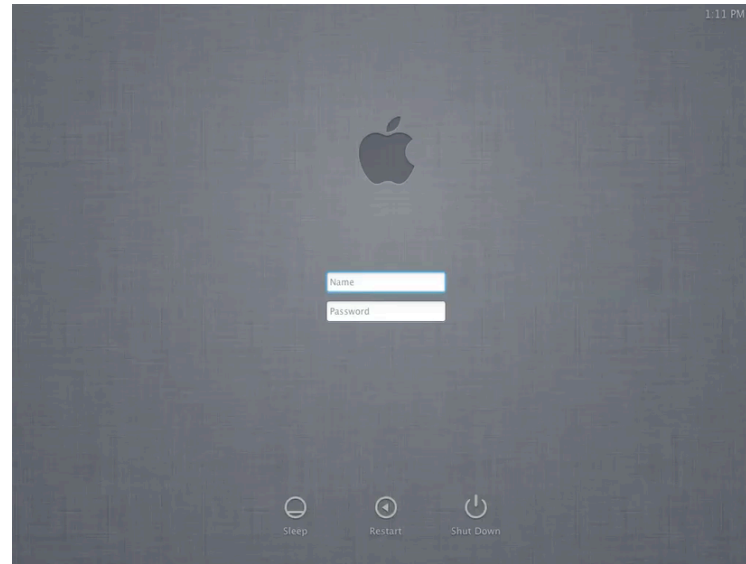
Recovery Key Type:

FileVault Enabled Users: Management Account
 Current or Next User

fdsetup enable -defer plistfile.plist

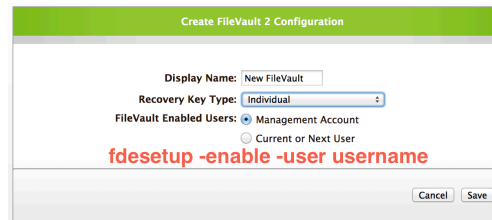
In the chosen Disk Encryption Configuration, since it's specifying that the current or next user will be enabled and that the individual recovery key is being used, the Casper agent will be running the `fdsetup enable -defer plistfile.plist` command shown on the screen.

Encryption Policies in Self Service



Let's take a look at how this will appear from the user's end.

Encryption Policies in Self Service



Create FileVault 2 Configuration

Display Name: New FileVault

Recovery Key Type: Individual

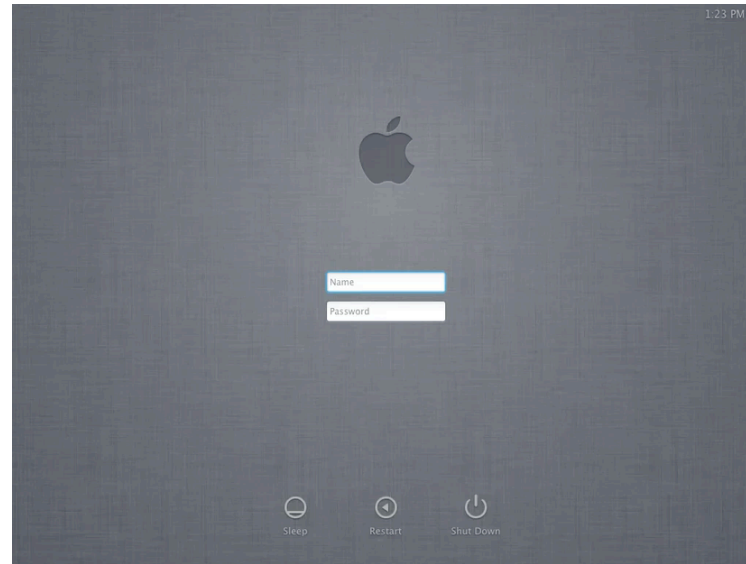
FileVault Enabled Users: Management Account
 Current or Next User

fdsetup -enable -user username

Cancel Save

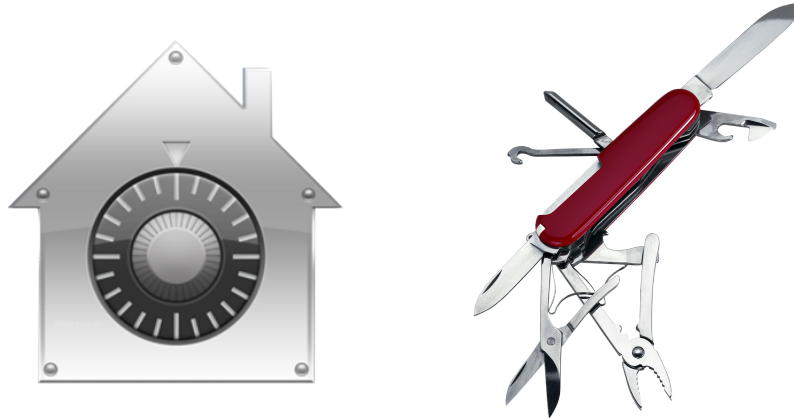
Let's take that same policy and change the Disk Encryption Configuration to have it specify that the Casper management account be used. With the user change, the Casper agent will be running the `fdsetup` command shown on the screen. I'll also be changing the displayed message, so that it requests a restart instead of a logout.

Encryption Policies in Self Service



With the new user info, let's take a look at how this updated policy will appear from the user's end.

`fdsetup` = FileVault 2 multi-tool



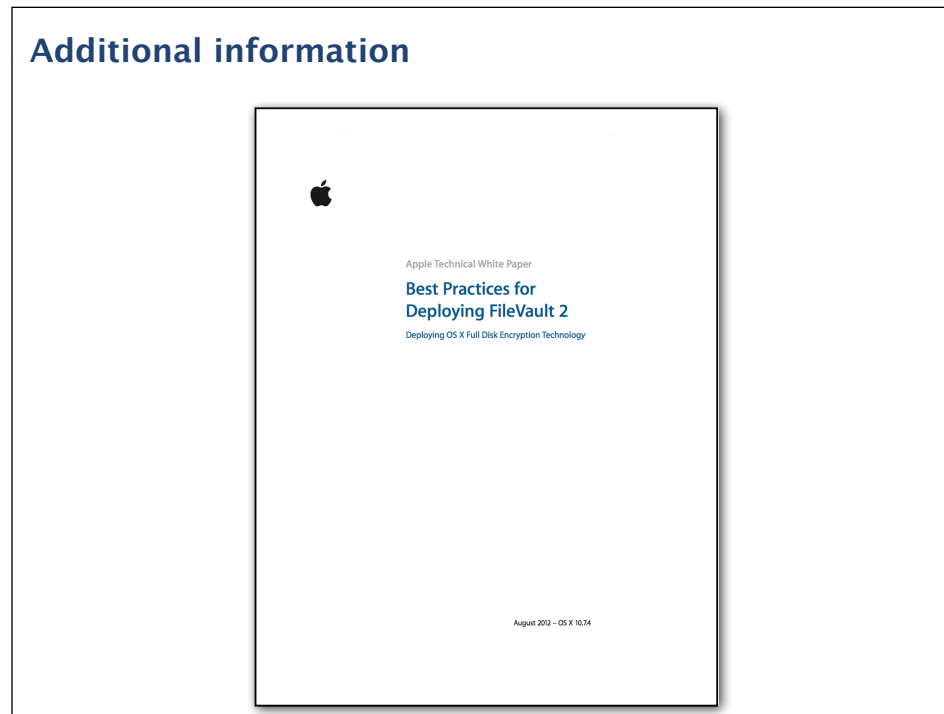
To wrap up, `fdsetup` is a Swiss Army knife for managing FileVault 2 on 10.8. It can enable FileVault 2, add and remove users, report on FileVault 2's status and more. If you're managing FileVault 2 encryption in your own environment, I recommend using this tool. Properly used, it will save you time and give encryption options available with no other software.

Additional information



If you want more information about fdesetup, I recommend checking out the July 2012 issue of MacTech. It's available via the MacTech iPad app and print copies should be available for ordering.

Additional information



For more general information on FileVault 2, Apple has put out a white paper that describes best practices for deploying FileVault 2 on Lion. Because it is focused on Lion, it does not cover `fdsetup` but it does include a lot of interesting technical detail on how FileVault 2 works.

Links

- › Apple Best Practices for Deploying FileVault 2 - <http://training.apple.com/osx>
- › Administering FileVault 2 on OS X Mountain Lion with the Casper Suite - <http://www.jamfsoftware.com/resources/white-papers>
- › Using fdesetup with Mountain Lion's FileVault 2 - <http://derflounder.wordpress.com/2012/07/25/using-fdesetup-with-mountain-lions-filevault-2/>
- › Embedding certificate data into a fdesetup plist file - <http://derflounder.wordpress.com/2012/08/22/embedding-certificate-data-into-a-fdesetup-plist-file/>
- › Encrypting Volumes in OS X Mountain Lion - <http://krypted.com/mac-os-x/encrypting-os-x-mountain-lion/>

Here's some useful links for FileVault 2 and fdesetup, including links to some topics not discussed as part of today's talk.

**PDF available from the
following link:**

<http://tinyurl.com/PSUMac2013PDF>

**Keynote slides available
from the following link:**

<http://tinyurl.com/PSUMac2013key>

As mentioned earlier, here are the download links for this talk. It's available in PDF format with the speakers notes and you can also download the Keynote slides with all of the speakers notes and demos.

Thank you for attending!

Enjoy the rest of the
conference

@rtrouton

FOLLOW ME ON 